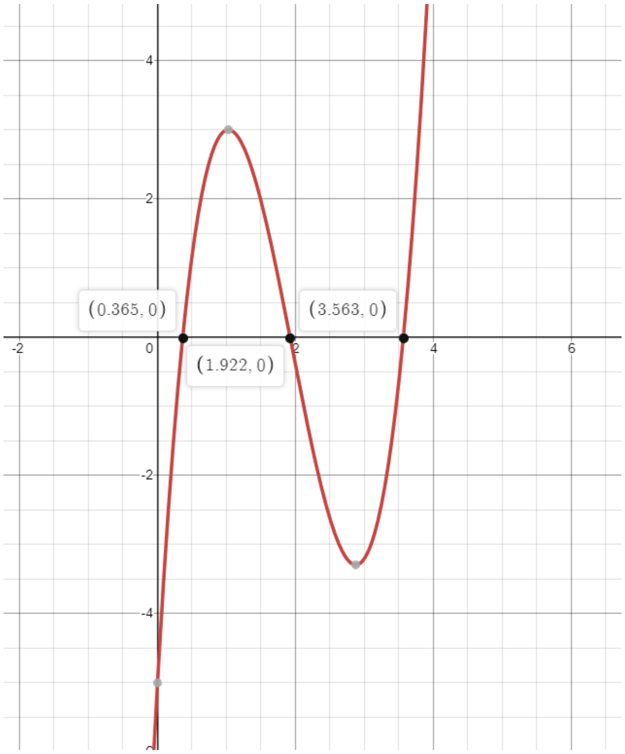
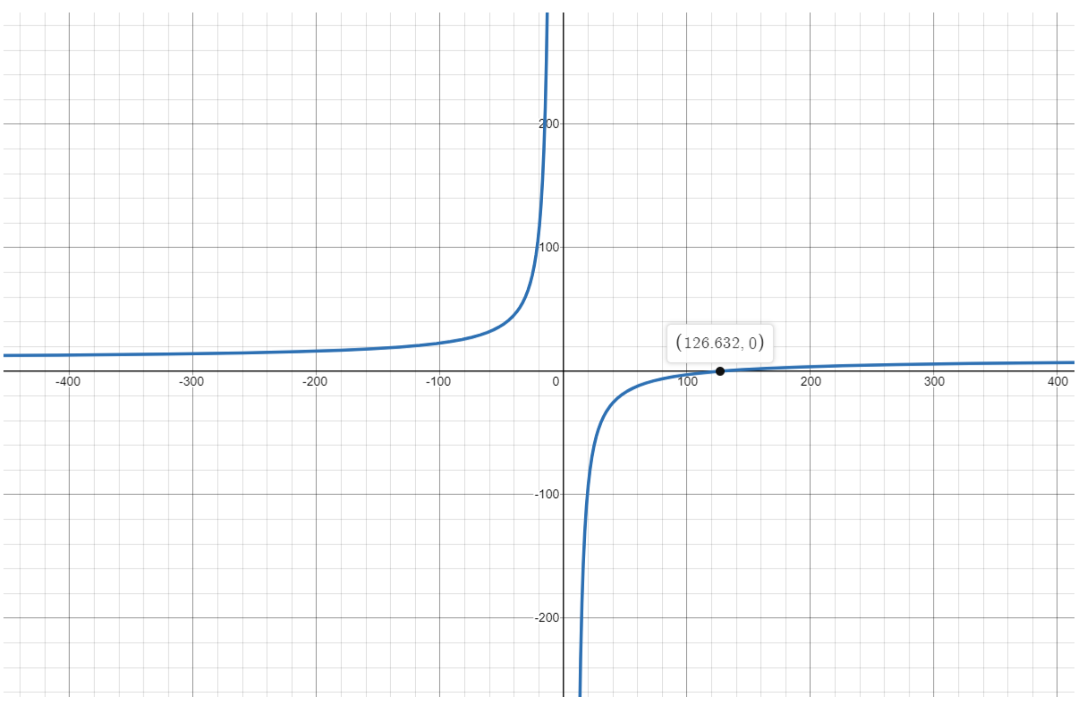
Graph of Functions

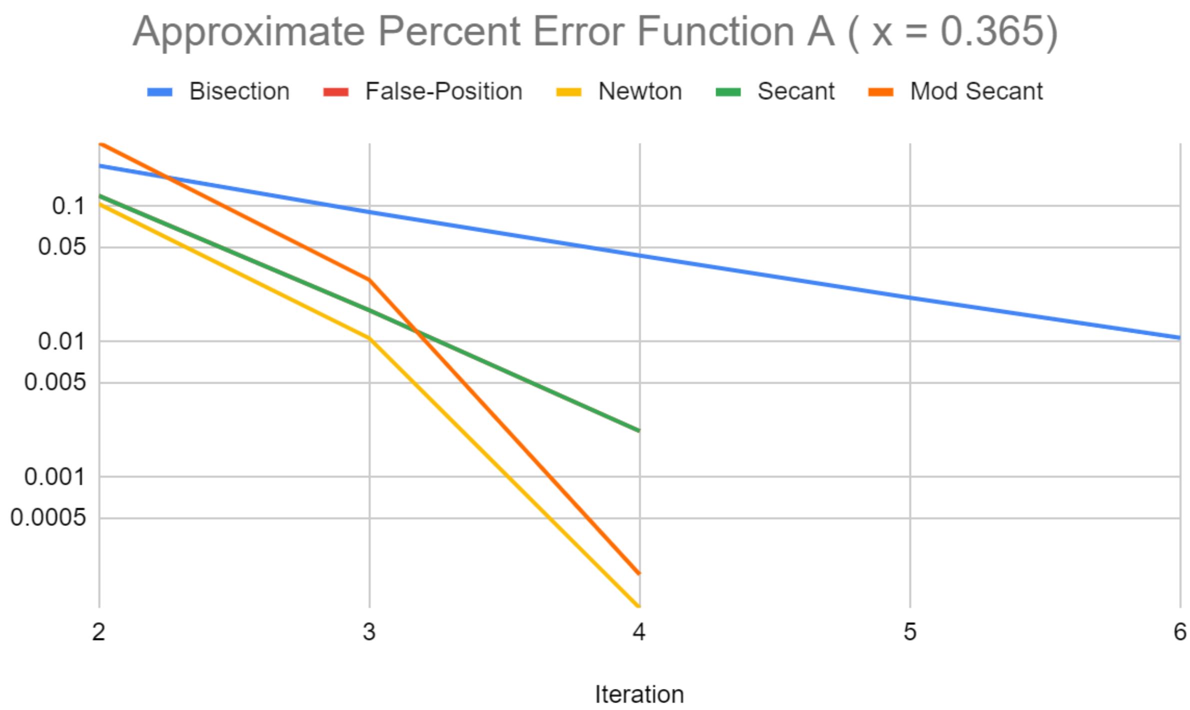
(a) $F(x) = 2x^3 - 11.7x^2 + 17.7x - 5$

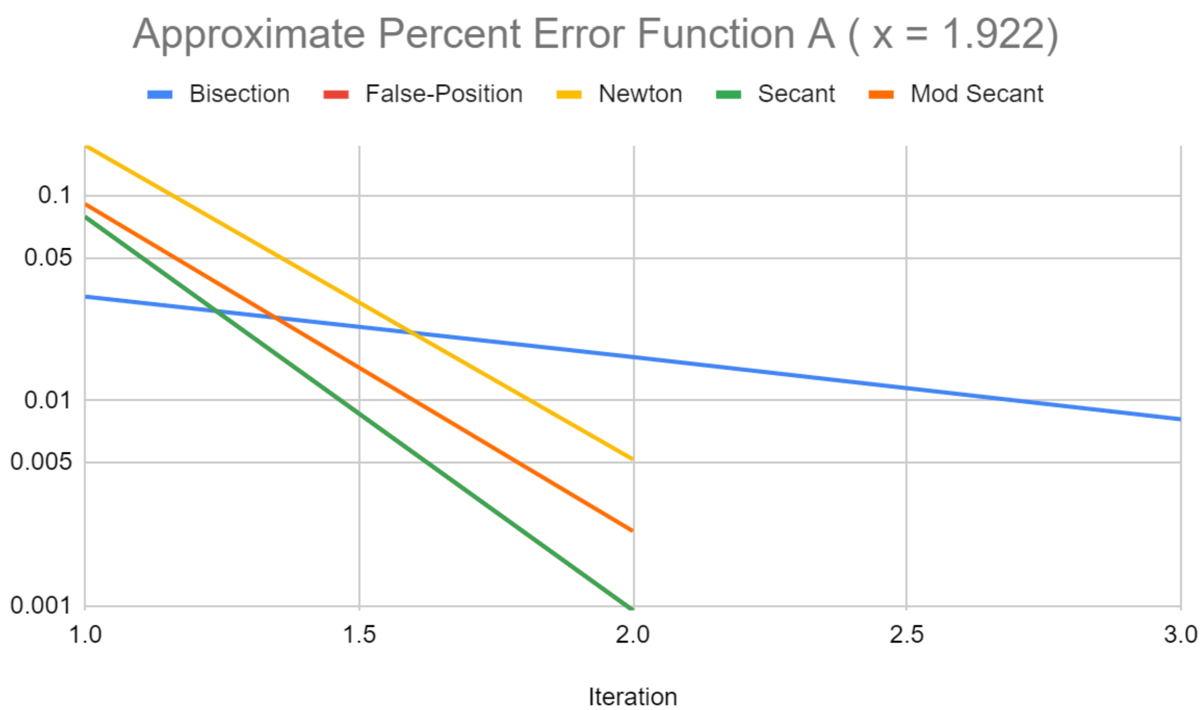


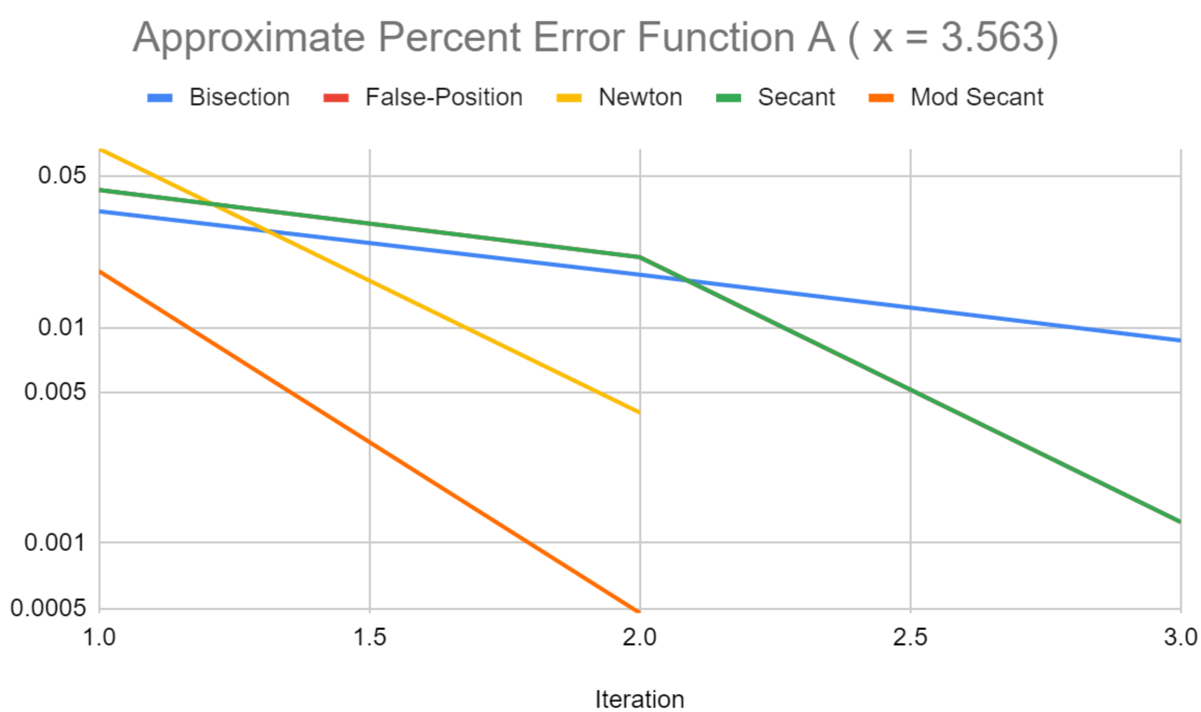
(b) $F(x) = x + 10 - x\cosh(50/x)$

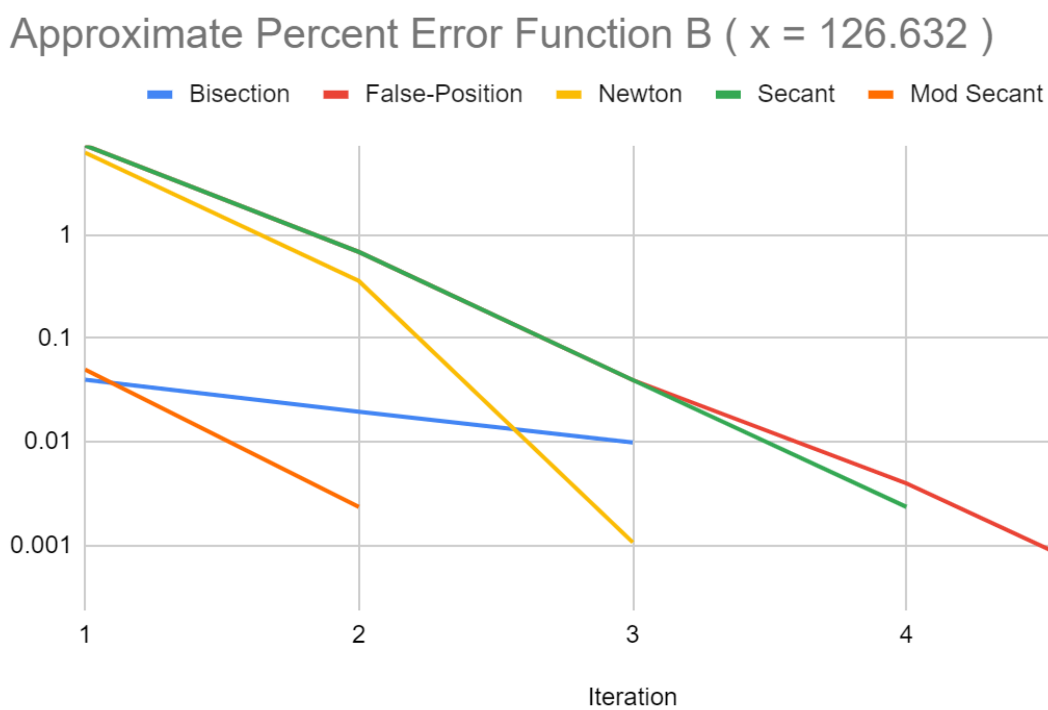


Approximate Percent Error




Approximate Percent Error Function A (x = 1.922)
Bisection
False-Position
Newton
Secant
Mod Secant
0.1
0.05
0.01
0.005
0.001
1.0
1.5
2.0
2.5
3.0
Iteration


Approximate Percent Error Function A (x = 3.563)
Bisection
False-Position
Newton
Secant
Mod Secant
0.05
0.01
0.005
0.001
0.0005
1.0
1.5
2.0
2.5
3.0
Iteration



There was no true root given in the problem, so I didn't include the true percent relative error

For Function A, I used points that are close to the root just so the number of iterations isn't too large. An issue occurred; however, because the False-Position and Secant error lines happen to overlap due to those methods essentially using the same technique when the chosen points bracket the root. All the methods generally finished around the same number of iterations except for the Bisection Method (which is the obviously the slowest). A strange thing to note is that according to the graphs, the Bisection method is also the least "accurate" in terms of approximate percent error. While the method did converge to the correct root, the error is significantly larger than the rest

For Function B, the story is a bit different. I used the points given in the assignment (bracketed between 120-140) and for some reason the Secant Method and False-Position method, while having the same general slope for the first three iterations branch off immediately after the third. Perhaps this is an error in my code; however, the results for Function A are accurate (or so it seems to me) so I think it may be a special case for the two methods. Its also surprising that the False-Position Method took the longest while the Bisection Method was tied for 2nd/3rd.